International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Design of Low POWER CLA using Coding-Based Partial MRF Method

M.Sravanthi[1], Dr.Ch.Venugopal Reddy[2]
PG-Student [1] & Professor&HoD[2]
Department of Electronics and Communication Engineering
RISE Krishna Sai Prakasam Group of Institutions (Autonomous), Ongole, AP, India
(sravanthyriseclg@gmail.com[1], phdvenu@gmail.com[2])

1. ABSTRACT:
One of the biggest challenges in designing circuits with low power consumption is reliability. It is common practice to employ Markov Random Field (MRF) approaches to deal with random noise in circuits operating at very low voltages. For smaller, simpler circuits, these function well, but for larger, more intricate designs, they become challenging. This work presents a novel method known as coding-based partial MRF (CPMRF). The CPMRF pair, which utilizes a shared MRF network to conserve space, is a unit that combines several logic processes. Additionally, by employing a coding technique that reinforces "1s" and "0s," it increases resistance to noise. The design is therefore more economical and efficient than previous approaches. The Cadence tool is used in this project to develop a Carry-look ahead adder (CLA) using CPMRF pairs. Measurement results show that, in comparison to the master-and-slave MRF CLA design, the CPMRF CLA can achieve high noise tolerance with a 20% improvement while taking up 37.7% less space and using 93% less power.

1.1. INTRODUCTION:

Power-efficient and noise-immune circuit design is becoming more and more difficult as semiconductor technology advances to the Nanoscale zone. Modern integrated circuits' (ICs') overall performance and dependability are impacted by the shrinking feature size, which makes them more vulnerable to noise, process fluctuations, and power dissipation problems. While useful, traditional low-power design strategies frequently fall short of addressing these issues holistically, requiring creative solutions to improve power efficiency and noise immunity.

The detecting technique often requires the availability of original data in architecture. Utilizing statistical modeling to optimize circuit behavior under various noise situations, the Coding-Based Partial Markov Random Field (MRF) Method is one such promising technique. Often utilized in probabilistic inference and image processing, the MRF framework is modified for Nanoscale circuit design to increase resilience.

Against external sources of noise and fluctuations in the process. An excellent option for ultra-low-power applications, the suggested approach further enhances error resilience and reduces energy usage by integrating coding techniques.

This journal examines how the Coding-Based Partial MRF Method is applied in Nanoscale circuit design, highlighting how it improves overall performance, reduces power consumption, and reduces noise. Through simulations and experimental confirmation, the methodology's potential in next-generation low-power electronic devices is demonstrated.

Noise immunity and power consumption are important considerations in the field of Nanoscale circuit design. The performance and dependability of electronics can be greatly impacted by noise and power leakage, which increases as they go smaller to the Nanoscale. Several approaches have been investigated by researchers to tackle these issues, such as partial Markov random field (MRF) methods based on coding.

Using the concepts of MRFs and coding theory, the coding-based partial MRF method is a new technique for designing low-power, noise-immune Nanoscale circuits. Probabilistic graphical models, or MRFs, are a useful tool for illustrating the relationships between various variables in a system. MRFs are a useful tool for modeling the behavior of circuit elements and their interactions in circuit design.

By splitting the MRF clique energy into two pieces, which correspond to logic 0 and 1 output, the partial MRF method is applied. As per the coding scheme, this permits the utilization of a mix of strong 0s and 1s. By allowing many logic functions to share a common MRF network, this method can drastically lower the circuit's space and power usage.

1.2. OBJECTIVE:

Designing and implementing a low-power, noise-immune Nanoscale circuit employing a partial MRF (Markov Random Field) method based on coding is the main objective of this research. The goals include creating an effective circuit design that combines coding techniques and statistical modeling based on MRF for increased error resilience.

- Reducing power loss while preserving circuit functionality.
- Improving noise resistance to external disruptions and process changes.
- Using simulations and performance analysis to assess the suggested design's efficacy.

1.3. Scope of Work:

This study investigates noise reduction techniques based on coding and focuses on digital circuit design using Nanoscale technology.

2. LITERATURE SURVEY:

TOPIC: Extensive Analysis of Noise-Immune, Low-Power Nanoscale Circuit Architecture

AUTHOR:

S. Bhanja, K. Lingasubramanian, and T. Rejimon (2009) Nano Domain Logic Circuits: Probabilistic Error Modeling Deep submicron technology's study and optimization of connection noise are covered by M. A. Elgamel and M. A. Bayoumi (2003). Complex digital systems are where substrate noise creation takes place, according to M.V. Heijningen et al. (2002). Scaling Trends of On-Chip Power Distribution Noise (A.V. Mezhiba and E. G. Friedman, 2004).

Reducing Crosstalk Noise in Synthesized Digital Logic Circuits (O. Milter and A. Kolodny, 2003) Analysis of Power Plane Resonance-Induced Ground Bounce (S. Van den Berghe et al., 1998) Reducing Switching Noise in a Power Distribution System(T.-H.Chang,2005)

Models of Compact Noise for MOSFETs (R. P. Jindal,2006) Examining Noise for Effective Energy Design of Deep Submicron VLSI Chips (W. Jun, 2005)

1.4. BENFITS:

- The power consumption of Nanoscale circuits can be considerably decreased using the MRF technique. Minimal level of computational complexity.
- These circuits' noise immunity can be enhanced, increasing their resistance to interference and noise.
- By reducing the space needed to install these circuits, smaller and more effective devices may result.It can reduce the area required to implement these circuits, which can lead to smaller and more efficient devices.
- Provides a novel NR strategy by utilizing deep learning techniques. It addresses one of the main gaps in No Reference P techniques for guaranteeing 3D mesh quality.
- It addresses one of the main gaps in No Reference P techniques for guaranteeing 3D mesh quality.

3. EXSTING METHOD

3.1 TYPES OF ADDER

3.1.1 HALF ADDER:

A half adder adds two one-bit binary numbers, A and B. It produces two outputs: the sum (S) and the carry (C), which is theoretically propagated to the next stage. The resulting value can be expressed as 2C+S2C + S2C+S.

Fig 3.1.1 Logical diagram of Half Adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 3.1.1: Truth Table of Half Adder

3.1.2 FULL ADDER

A full adder performs binary addition while also considering both the input and output carry values. It adds three one-bit binary inputs typically labeled A, B, and Cin where A and B are the primary operands and Cin represents the carry input from the previous, less significant stage.

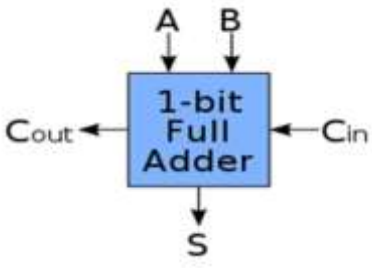


Fig 3.1.2 Schematic symbol for a 1-bit full adder

3.1.3 RIPPLE CARRY ADDERS (RCA)

An N-bit Ripple Carry Adder (RCA) is constructed by connecting N full adders in series.

In this configuration, the carry-out from each full adder serves as the carry-in for the next stage. The sum and carry outputs are determined using the following equations:

Si = Ai xor Bi xorCi Ci+1 = Ai Bi + (Ai + Bi) Ci

Since the carry must propagate through each full adder sequentially, the total delay corresponds to the longest critical path, resulting in a worst-case delay scenario. Although the RCA is the slowest among all types of adders with a time complexity of O(n), it is highly area-efficient with a space complexity of O(n). When implemented using N full adders, the propagation delay from input carry Cin to output carry Cout amounts to approximately 2N gate delays. Consequently, the overall delay increases linearly with the number of bits. The block diagram of the RCA is illustrated in Figure 3.1.4.

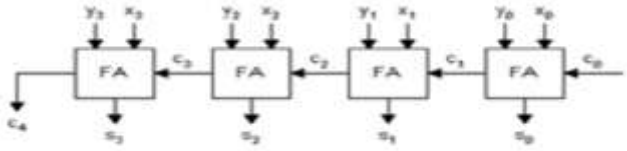


Fig 3.1.3: Block diagram of Ripple Carry Adder (RCA)

3.4. CARRY-LOOKAHEAD ADDER (CLA)

A particular kind of adder used in digital logic is called a carry-look ahead adder (CLA). Carry-look ahead adders increase speed by cutting down on the time needed to figure out carry bits. It can be compared to a ripple carry adder, which is simpler but typically slower. In this adder, the carry bit is calculated along with the sum bit, and each bit must wait for the previous carry to be calculated before it can start calculating its own result and carry bits (for more information; see adder). In order to minimize the wait time for calculating the result of the larger value bits, the carry-look ahead adder computes one or more carry bits prior to the total. The Brent-Kung adder and the KoggeStone adder are two instances of this kind of adder.

The operation of a ripple-carry adder is identical to that of pencil-and-paper methods. A result is derived by adding the two corresponding digits, beginning at the rightmost (least significant) digit position.

Two factors determine carry view ahead: 1) Determining if a carry will be propagated at each digit place if one enters from the right. 2) Combining these computed values to rapidly determine if a given set of digits will propagate a carry that enters from the right.

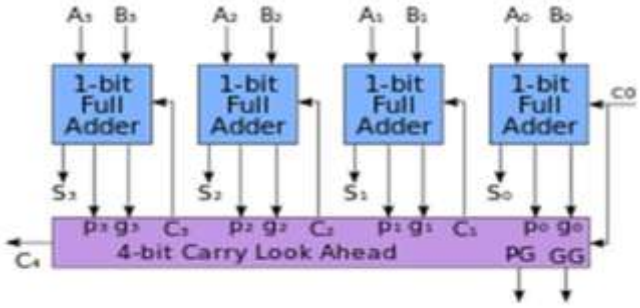


Fig.3.1.4. Carry Look Ahead Adder (CLA)

The logic behind the propagate (p) and generate (g)values in the given example is presented below. Observe that the signal from the circuit above is determined by the numerical value, which ranges from 0 on the far left to 3 on the far right:

$C_1=G_0+ P_0. C_0$

$C_2=G_1+ P_1. C_1$

$C_3=G_2+ P_2. C_2$

$C_4=G_3+ P_3. C_3$

Substituting C1into C2, then C2 into C3, then C3 into C4yields the expanded equations:

$C_1=G_0+ P_0. C_0$

$C_2=G_1+ G_0. P_1. +C_0 .P_0. P_1$

$C_3=G_2+ G_1.P_2 +G_0. P_1. P_2+ C_0 .P_0. P_1. P_2$

$C_4=G_3+ G_2.P_3 +G_1. P_2. P_3+ C_0 .P_0. P_1. P_2. P_3$

To determine whether a bit pair will generate a carry, the following logic works:

$$G_i=A_i. B_i$$

One of the logic statements below can be used to determine if a bit pair will propagate a carry:

$$P_i=A_i \oplus B_i$$

$$P_i=A_i + B_i$$

3.1.5. CARRY SAVE ADDER

A carry-save adder is a kind of digital adder that is used in computer micro architecture to calculate the binary sum of three or more n-bit values. In contrast to conventional digital adders, it produces two numbers with the same dimensions as the inputs: a carry bit sequence and a partial sum bit sequence. Take the following sum into consideration: 12345678+87654322=100000000.

To calculate the total from right to left, we use simple arithmetic: "8+2=0, carry 1", "7+2+1=0, carry 1", "6+3+1=0, carry 1", and so forth. Even though we can immediately tell the result's last digit, we cannot tell the

first digit until we have passed the carry from each digit to the one on its left after going through every digit in the computation. Therefore, even though the equipment we're using might execute multiple calculations at once, adding two n-digit values must take a time proportionate to n.

Delays can be minimized with a carry look-ahead adder. The delay can theoretically be decreased to be proportional to log n, but for large numbers, this is not possible because propagation delays increase at the same rate as the distances that signals must travel on the chip, even when carry look-ahead is used. Carry look-ahead becomes ineffective as we reach the 512-bit to 2048-bit number sizes needed for public-key cryptography.

3.2 DRAWBACKS

1. At every point in a carry-save addition, we are immediately aware of the outcome.
2. We are still unsure if the addition's result is greater or less than a specified number (for example, we are unsure if it is positive or negative).

A series of Montgomery multiplications saves time, but a single one does not. Montgomery multiplication, which is dependent on the rightmost digit of the result, is one solution; nevertheless, it has a fixed overhead, similar to carry-save addition itself.

Fortunately, exponentiation is the most widely used operation in public-key cryptography. It is essentially a series of multiplications.

N complete adders make up the carry-save unit, and each one uses only the corresponding bits of the three input values to calculate a single sum and carry bit. A partial sum ps and a shift-carry sc are generated given the three n-bit values a, b, and c:

$PS_i = a_i \oplus_i \oplus$

$SC_i = (a_i \wedge b_i)\ v\ (a_i \wedge c_i)\ v\ (b_i \wedge c_i)$

The entire sum can then be computed by:

1. Shifting the carry sequence sc left by one place.
2. Appending a 0 to the front (most significant bit) of the partial sum sequence ps.
3. Using a ripple carry adder to add these two together and produce the resulting n + 1-bit value.

3.3 MRF-BASED CIRCUIT DESIGN METHODOLOGY

Based on the MRF hypothesis, the MRF circuit

$$P(X) = \prod_{c \in C} \frac{1}{Z} \exp\left(\frac{-U_c(x_c)}{k_b T}\right) \quad (1)$$

design was created. An MRF network is mapped to MRF standard cells once a multi-level Boolean logic circuit has been mapped to an MRF network. The mapping procedures used in the MRF design as well as some fundamental ideas will be covered in this part.

3.4 Markov Random Field Theory

Let X = {x0, x1,...,xn} be a set of random variables. A graph example is shown in Fig. 1,

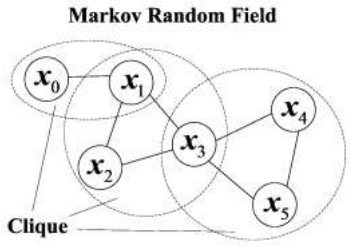


Fig.3.4.1 Graph example of MRF

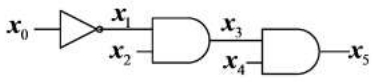


Fig.3.4.2 Logic circuit.

The values of X are {x0, x1,..., x5}. Direct or indirect relationships exist between these variables. The term clique refers to the subset of X where variables are directly related to one another [24]. {x0, x1}, {x1, x2, x3}, and {x3, x4, x5} are cliques in Fig. 1. In Figure 1, the variables in a clique exhibit clear dependence, as seen by the edges connecting two variables. A neighbor is a directly related variable that is next to another variable in these subsets [24]. A neighbor of x1 is x0. Only an MRF variable's neighborhood affects its likelihood. If the sets of random variables meet positivity (a variable's probability cannot be negative or zero) and markovianity (a variable must be independent of all other variables save those in its neighborhood), then these variables form an MRF.

3.5 Mapping a CMOS-Circuit to an MRF Network

A circuit's joint probability for all of its nodes is equal to the product of all clique energy functions, as per the Hammersley–Clifford theorem.

$$P(X) = \prod_{c \in C} \frac{1}{Z} \exp\left(\frac{-U_c(x_c)}{k_b T}\right) \quad (1)$$

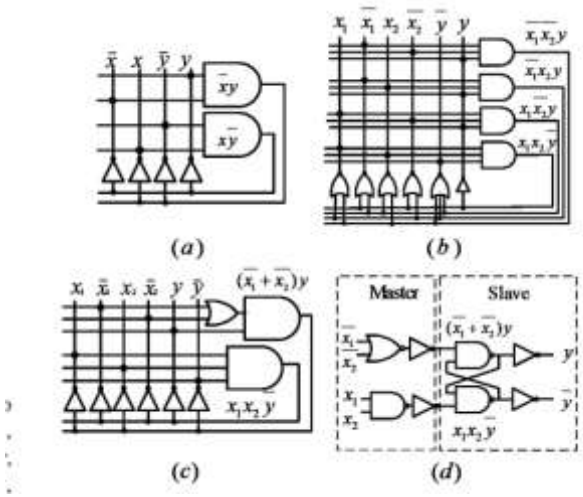


Fig.3.5 MRF circuits (a) MRF inverter (b) MRF NAND (c) Area-efficient MRF NAND
(d) MS MRF NAND

Energy Truth Table of an Inverter

Input	Output	State	Clique Energy (x,y) = -(
0	0	Incorrect/Invali	0
0	1	Correct/Valid	-1
1	0	Correct/Valid	-1
1	1	Incorrect/Invali	0

Table.3.5

3.6Mapping an MRF Network onto an MRF-Based Circuit

High-performance fundamental MRF logic gates are made to map an MRF network into an MRF-based circuit.

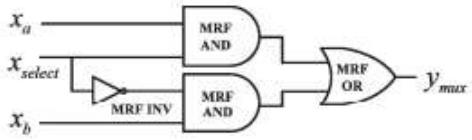


Fig. 3.6.1Local mapping-based MRF MUX

The MRF-NAND is a direct hardware implementation that uses the clique energy in the manner described below.

$$U(x_1, x_2, y) = -(\bar{x}_1\bar{x}_2y + \bar{x}_1x_2y + x_1\bar{x}_2y + x_1x_2\bar{y}). \quad (2)$$

The corresponding transformation of the clique energy function is used to modify the MRF-NAND structure.

$U(x_1, x_2, y) = -[(\bar{x}_1.\bar{x}_2)y + x_1.x_2.\bar{y}]$ (3)

3.7 MRF carry-look ahead adder

To increase addition speed, a conventional Carry Look-Ahead Adder (CLA) recomputed carry signals instead of relying on ripple carry propagation. to further enhance carry computation, reduce power consumption, and mitigate noise sensitivity in deep-submicron and Nanoscale VLSI implementations, the Markov Random Field (MRF) technique employs probabilistic modelling.

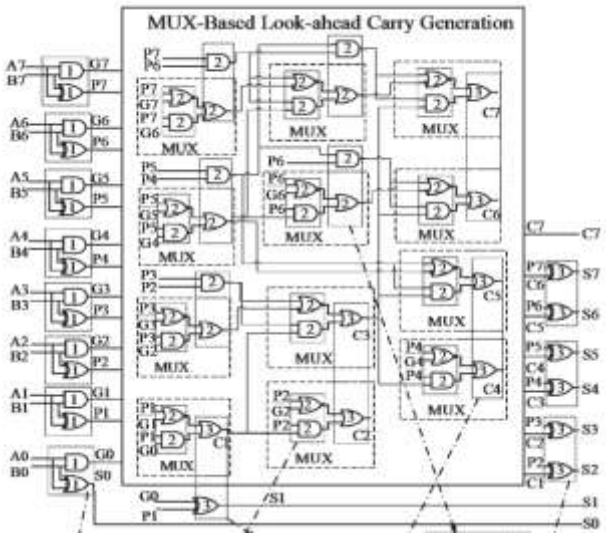


Fig.3.7MUX-Based Look -Ahead Carry Generation

4. ADVANCED METHOD

The partial MRF methodology based on coding

The sub-clique energy is the basis for a suggested CPMRF approach in cases where the clique energy is asymmetric. The first example of a complimentary pair is a PMRF AND–NOR pair. Based on the AND–NOR pair's truth table in Table II, the partial clique energy Up of AND is

$U_p{-}AND(x_1, x_2, y) = -\bar{x}_1.x_2.\bar{y} - x_1.\bar{x}_2.\bar{y} - \bar{x}_1.\bar{x}_2.\bar{y}$ (4)

In contrast, NOR's partial clique energy is

$U_p{-}NOR(x_1, x_2, y) = -\bar{x}_1.x_2.\bar{y} - x_1.\bar{x}_2.\bar{y} - x_1.x_2.\bar{y}$

$=-(x_1+x_2).\bar{y}$ (5)

One of the $U_P{-}AND$ phrases, x1 • x2 • y $_{AND}$, might be used to symbolize the lost clique energy of NOR.

$U_{NOR=}\ U_p{-}NOR - \bar{x}_1.\bar{x}_2.\bar{y}_{AND}$ (6)

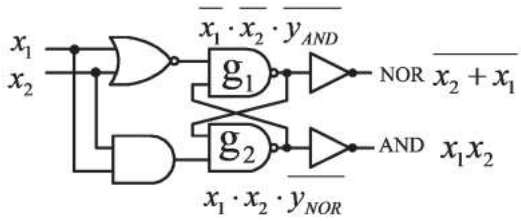


Fig.4.1.1 Compensation clique energy-based structure for an AND–NOR pair.

One of the UP−NOR terms x1 · x2 · y_{NOR} to represent the lost clique energy of AND

$U_{AND=}$ Up-AND-$x_1.x_2.\bar{y}_{NOR}$ (7)

TRUTH TABLE OF AN AND–NOR PAIR

Input			Valid output			
x_1	x_2		AND y_{AND}		NOR y_{NOR}	
0	0	0	U_{p-AND}	$\bar{x}_1 \cdot \bar{x}_2 \cdot \overline{y_{AND}}$	U_1	1
0	1	0		$\bar{x}_1 \cdot x_2 \cdot \overline{y_{AND}}$	$\bar{x}_1 \cdot x_2 \cdot \overline{y_{NOR}}$	0
1	0	0		$x_1 \cdot \bar{x}_2 \cdot \overline{y_{AND}}$	$x_1 \cdot \bar{x}_2 \cdot \overline{y_{NOR}}$	0
1	1	1	U_1	$x_1 \cdot x_2 \cdot \overline{y_{NOR}}$		0

Table.4.1

The clique energy to represent the lost energy terms of an AND–NOR pair is

U_c-AND-NOR = $-x_1 .x_2.\bar{y}_{NOR}- \bar{x}_1. \bar{x}_2. \bar{y}_{AND}$ (8)

In equation (8), the lost clique energy of AND in (7) and the lost clique energy of NOR in (6) are simply added. Whereas the phrase x1 • x2 • yNOR of NOR may be thought of as a compensation for the lost clique energy of AND, the term x1 • x2 • yAND of AND can likewise be thought of as a compensation for the missed clique energy of NOR in (8). Consequently, we use (6)–(8) to simplify the joint clique energy of an AND–NOR pair, as seen in Table

$U_{AND-NOR} = -(x_1 .x_2.\bar{y}_{NOR}+ \bar{x}_1. \bar{x}_2. \bar{y}_{AND})$

$- \bar{x}_1 .x_2.\bar{y}_{NOR}+ x_1 . \bar{x}_2.\bar{y}_{NOR}$

$- \bar{x}_1 .x_2.\bar{y}_{AND}+ x_1 . \bar{x}_2.\bar{y}_{AND}$

$= U_{c-AND-NOR}-(x_1 \oplus). \bar{y}_{NOR}$

$-(x_1 \oplus). \bar{y}_{AND}$

As illustrated in Figure 6, gates g4 and g5 are suggested for the corresponding clique energy of the function UC−AND−NOR.

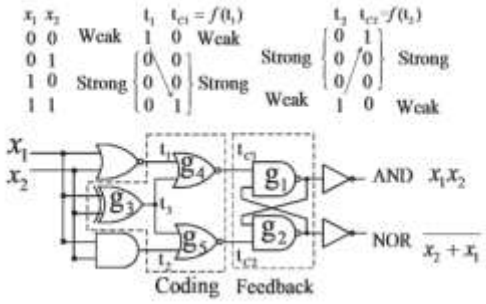


Fig.4.1.2Coding-based structure for a complementary AND–NOR pair with two logic operations.

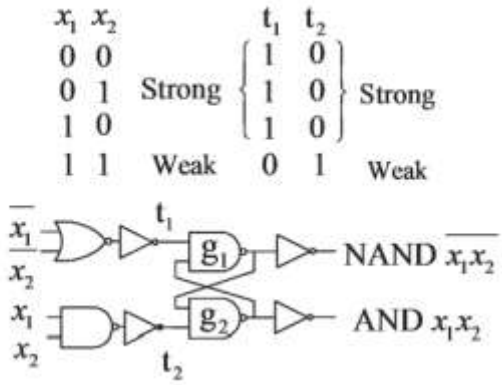


Fig. 4.1.3MS [28] structure with only one kind of logic

When noise is present, the likelihood of an input being weak is pw($0 \le pw < 0.5$), and the probability of an input being strong is 1− pw. For instance, a NAND gate has x1, x2, and y as its inputs and y as its output. The conditional probability that output y is strong is represented by the notation p(y|x1x2).

$$
\begin{aligned}
p(1|00) &= 1 - p_w^2 \\
p(1|01) &= 1 + p_w^2 - p_w \\
p(1|10) &= 1 + p_w^2 - p_w \\
p(0|11) &= (1 - p_w)^2 \\
p(1|00) &\ge p(1|01) = p(1|10) \ge p(0|11). \qquad (10)
\end{aligned}
$$

The suggested structure can withstand more noise than traditional MRF structures. The AND-XOR and AND-NOR examples above suggest that a great number of complimentary modules are feasible: The self-complementary pairs NAND–OR, AND–XOR (XNOR), OR–XOR (XNOR), NAND–XOR (XNOR), and NOR–XOR (XNOR) are created by logic transformation. It is possible for two logic operations to share a common MRF network within the same complimentary module.

This case study uses the following transformation to expand the two inputs to three-input AND-NOR pairings for the MUX function:

$$y_{mux} = x_a \cdot x_s + x_b \cdot \overline{x_s} = x_a \cdot x_s + \overline{\overline{x_b} + x_s}. \qquad (11)$$

By the proposed coding-based AND–NOR

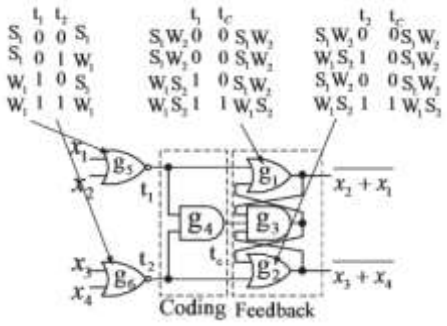


Fig. 4.1.4 Non-complementary CPMRF pair: NOR–NOR

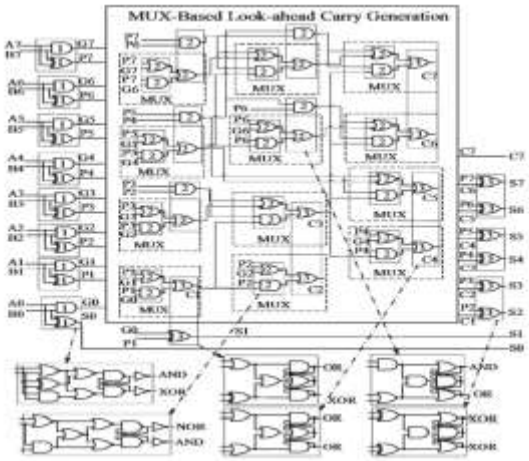


Fig.4.1.5.Proposed CPMRF carry-look ahead adder.

4.2 Essential Elements of the Partial MRF Approach

- Effective Computation of Residual Flow
- Reduced Computational Complexity
- Localized Updates
- Scalability
- A faster rate of convergence
- Application in Image Processing

4.3 Applications

1. Computer Vision & Image Processing
2. Natural Language Processing (NLP)
3. Machine Learning & Data Science
4. Finance & Economics
5. Biological & Medical Applications

5. SIMULATION OUTPUTS

5.1MRF inverter

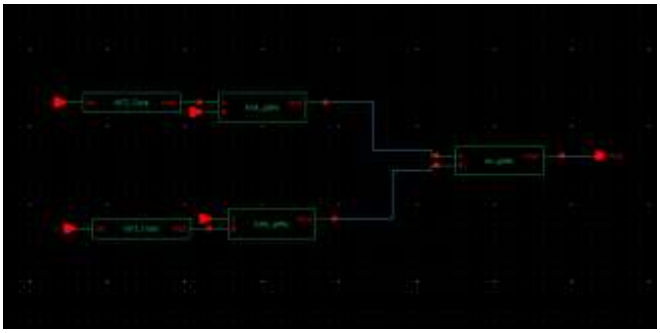


Fig.5.1MRF Inverter

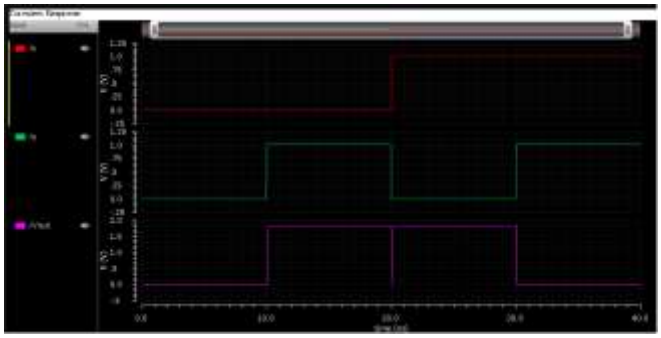


Fig. 5.2Simulation output of MRF NOT Gate

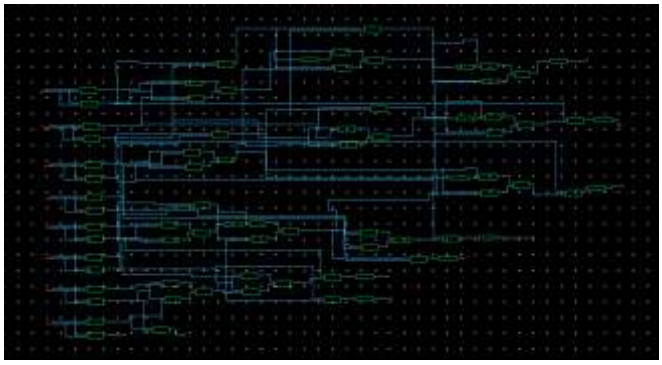
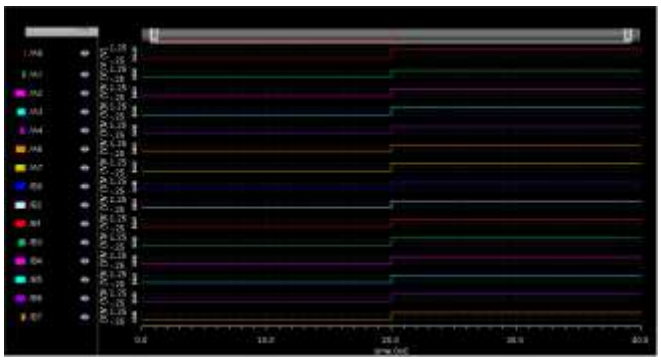


Fig.5.3 Carry Look Adder Design

Fig.5.4CLA Test Bench



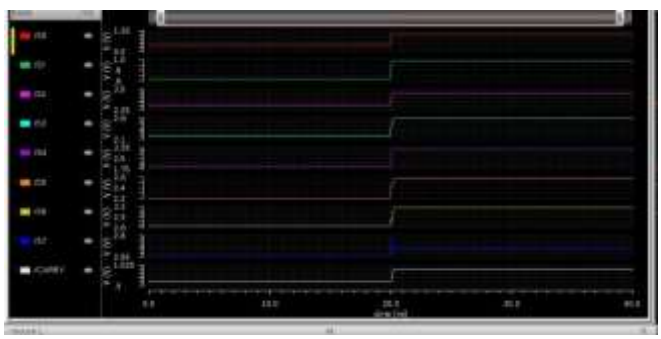


Fig.5.5 Simulation Output

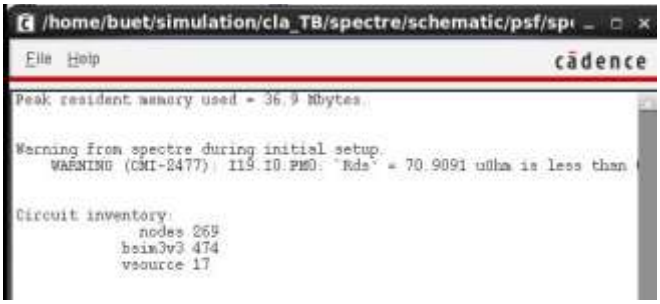


Fig.5.6 Area of CLA

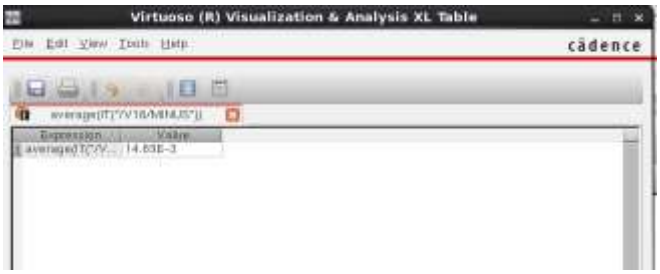
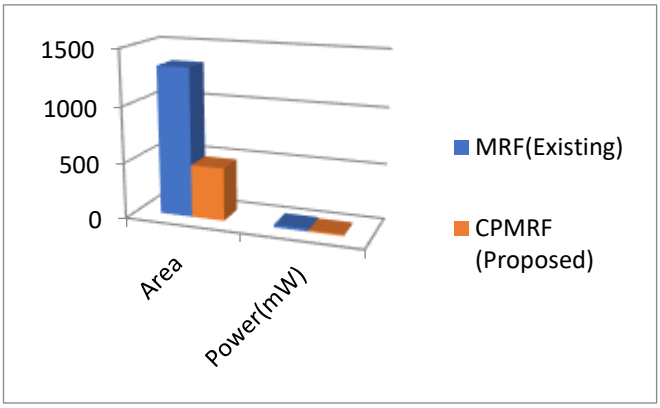


Fig.5.7 Power of CLA

Table-5.1 Compression of Existing and proposed Method

Parameter	Area	Power(mW)
MRF(Existing)	1332	24.606
CPMRF (Proposed)	475	14.659

Compression of Existing and proposed Method in Graphical View



CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The use of a coding-based partial Markov Random Field (MRF) approach for the design of low-power, noise-immune Nanoscale circuits has been investigated in this study, with a particular emphasis on its application in a carry-look ahead adder (CLA). Our objective has been to tackle the significant issues of noise susceptibility and power dissipation that afflict Nanoscale circuit design by utilizing the probabilistic modeling capabilities of MRFs and including error-correcting codes. Due to the constraints of conventional design techniques, the literature analysis emphasized the growing significance of these challenges as feature sizes continue to decrease.

Compared to traditional methods, our work has shown that the suggested methodology has the potential to make notable improvements in noise immunity and power usage. Effective circuit behavior modeling and the direct integration of coding techniques into the design process are made possible by the partial MRF framework. We have demonstrated through simulation and analysis of the MRF-based CLA how well the suggested approach works to improve adder performance, especially in noisy settings. The findings suggest that the partial MRF strategy based on coding presents a viable path towards the creation of more resilient and energy-efficient Nano scale arithmetic units, which are essential components of digital systems.

6.2 FUTURE SCOPE

The following intriguing directions for further research and development are laid out by this project: **Analysis of Complex Coding Schemes:** Though this effort focused on certain coding methods, future research into more complex codes—like LDPC codes, polar codes, or turbo codes—may result in even bigger gains in power economy and noise immunity. There should be more research done on the best coding strategies for various circuit layouts and noise levels. The creation of codes that are especially suited to the properties of MRF-based circuits may also prove to be a productive field of study.

Automated Design Flow: Making the suggested coding-based partial MRF method more practically applicable would be greatly improved by creating an automated design flow. For this, tools for circuit creation, code selection, performance optimization, and MRF model development would need to be developed. A toolchain like this would make it simple for designers to explore the design space and optimize circuits for certain performance goals.

Extension to Increasingly Complex Arithmetic Units: A CLA was the main emphasis of this effort. The wider usefulness of the coding-based partial MRF method would be demonstrated by extending it to more intricate arithmetic units, such as multipliers, divisions, and floating-point units. The potential advantages in terms of power and noise immunity are significant, but the construction of MRF models and energy reduction

are made more difficult by these more complicated circuits.

Validation and Hardware Implementation: In the end, hardware implementation is required to confirm the efficacy of any circuit design process. Testing and fabricating prototype circuits created with the suggested approach specifically, the MRF-based CLA and other arithmetic units would provide important details about how well it works in practice and point out any potential drawbacks. Additionally, this would make it possible to evaluate noise immunity and power usage more precisely.

Application to developing Technologies: The concepts of the coding-based partial MRF technique may find use in memristor-based circuits, spintronic devices,or quantum computing, among other developing technologies outside of CMOS. By investigating these options, new circuit design methods for upcoming computer paradigms may be developed.

Adaptive Noise Mitigation: Examining adaptive noise mitigation strategies in the context of the MRF framework may augment the resilience of the circuits that have been constructed. Adapting the circuit characteristics or coding scheme dynamically in response to the noise levels measured could be one way to do this. In circumstances that change quickly, such adaptable approaches could be quite beneficial.

Security Applications: Circuits could be protected from hardware Trojans and side-channel assaults by utilizing the natural noise immunity that coding techniques offer. Particularly in safety-critical systems, investigating these possible applications is an intriguing avenue for further study.

7. REFERENCES

[1] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 55–65, Jan. 2009.

[2] H. Elgamel and M. A. Bayoumi, "Interconnect noise analysis and optimization in deep submicron technology," *IEEE Circuits Syst. Mag.*, vol. 3, no. 4, pp. 6–17, 2003.

[3] M. V. Heijningen, M. Badaroglu, S. Donnay, G. G. E. Gielen, and H. J. D. Man, "Substrate noise generation in complex digital systems: Efficient modeling and simulation methodology and experimental verification," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1065–1072, Aug. 2002.

[4] A. V. Mezhiba and E. G. Friedman, "Scaling trends of on-chip power distribution noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 4, pp. 386–394, Apr. 2004.

[5] O. Milter and A. Kolodny, "Crosstalk noise reduction in synthesized digital logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 1153–1158, Dec. 2003.

[6] S. Van den Berghe, F. Olyslager, D. de Zutter, J. de Moerloose, and W. Temmerman, "An examination of ground bounce due to power plane resonances," *IEEE Trans. Electromagn. Compat.*, vol. 40, no. 2, pp. 111–119, May 1998.

[7] R. P. Jindal, "Compact noise models for MOSFETs," *IEEE Trans. Electron Devices*, vol. 53, no. 9, pp. 2051–2061, Sep. 2006.

[8] W. Jun, "Taking noise into account for effective energy design of deep submicron VLSI chips," in *Proc. Asia-Pacific Microw. Conf.*, Dec. 2005, p. 4.

[9] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ, USA: Princeton Univ. Press, 1956, pp. 43–98.

[10] J. Han, J. Gao, P. Jonker, Y. Qi, and J. A. B. Fortes, "Toward hardware redundant, fault-tolerant logic for nano electronics," *IEEE Des. Test Comput.*, vol. 22, no. 4, pp. 328–339, Jul. 2005.

[11] B. D. Brown and H. C. Card, "Stochastic neural computation: I. Computational elements," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[12] W. Qian, M. Riedel, H. Zhou, and J. Bruck, "Using combinational logic to transform probabilities," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 9, pp. 1279–1292, Sep. 2011.

[13] G. Sarkis and W. J. Gross, "Efficient stochastic decoding of non-binary LDPC codes with degree-two variable nodes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1279–1281, Dec. 2011.

[14] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5617–5626, Nov. 2011.

[15] Q. T. Dong, M. Arzel, C. Jego, and W. J. Gross, "Stochastic decoding of turbo codes," *IEEE Trans. Signal Process.* vol. 58, no. 12, pp. 6421–6425, Dec. 2010.

[16] X. Chen, H. Hu, and J. Sobelman, "An algorithm and hardware design for a stochastic iterative MIMO

detection system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 4, pp. 1205–1214, Apr. 2015.
[17] Y. Liu and K. K. Parhi, "Lattice FIR digital filter architectures using stochastic computing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 1027–1031.
[18] I.-C.Wey,C.-C. Peng, and F.-Y. Liao, "Reliable low-power multiplier design using fixed-width replica redundancy block," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 1, pp. 78–87, Jan. 2015.
[19] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 497–510, May 2004.